

# Unsupervised Total Variation Loss for Semi-supervised Deep Learning of Semantic Segmentation

Mehran Javanmardi, Mehdi Sajjadi, Ting Liu and Tolga Tasdizen  
`{mehran,mehdi,ting,tolga}@sci.utah.edu`

University of Utah, Salt Lake City, UT, 84112 USA

**Abstract.** We introduce a novel unsupervised loss function for learning semantic segmentation with deep convolutional neural nets (ConvNet) when densely labeled training images are not available. More specifically, the proposed loss function penalizes the  $L_1$ -norm of the gradient of the label probability vector image, *i.e.* total variation, produced by the ConvNet. This can be seen as a regularization term that promotes piecewise smoothness of the label probability vector image produced by the ConvNet during learning. The unsupervised loss function is combined with a supervised loss in a semi-supervised setting to learn ConvNets that can achieve high semantic segmentation accuracy even when only a tiny percentage of the pixels in the training images are labeled. We demonstrate significant improvements over the purely supervised setting in the Weizmann horse, Stanford background and Sift Flow datasets. Furthermore, we show that using the proposed piecewise smoothness constraint in the learning phase significantly outperforms post-processing results from a purely supervised approach with Markov Random Fields (MRF). Finally, we note that the framework we introduce is general and can be used to learn to label other types of structures such as curvilinear structures by modifying the unsupervised loss function accordingly.

**Keywords:** Semantic segmentation, Semi-supervised learning, Sparse labeling, Convolutional nets

## 1 Introduction

Semantic segmentation can be described as assigning an object label to each pixel in an image. This assignment can be binary, *i.e.* foreground vs. background or multi-class, *i.e.* "sky", "building", "horse" etc. Semantic segmentation is a challenging computer vision task that combines image segmentation and object detection, and can be seen as a crucial step for image understanding. Recently, ConvNets have been producing state-of-the-art results in many computer vision tasks including semantic segmentation [1,2]. However, successfully training ConvNets which can have large numbers of free parameters with existing methods requires large amounts of labeled data. Zhu *et al.* [3] claim that improvements have a correlation not only with the methodologies used but also the amount of

training data available to the algorithm. Unfortunately, building large labeled datasets can be costly and time consuming. At a time when access to unlabeled data is very easy and cheap, devising new algorithms and methods to profit from these sources is imperative. The goal of these algorithms can be to improve current state of the art results by incorporating more unlabeled data in a "*transductive learning*" setting or achieve results comparable to the state of the art with significantly smaller amounts of labeled training data in a "*semi-supervised learning*" paradigm which is the goal of this work.

Numerous methods and algorithms have been proposed for semi-supervised learning both in general and also for specific applications [4,5]. Self-training and co-training can be mentioned as classic methods in semi-supervised learning literature [6]. In this paper, we consider the problem of semantic segmentation when only sparse labels are available to the learning algorithm. This can be the case where a user labels a dataset only for some pixels per image which is a more realistic burden on the user than labeling all pixels in the training images. Motivated by this problem, we demonstrate how natural image characteristics, *i.e.* piecewise smoothness, can be used to develop novel unsupervised loss functions for learning ConvNets and demonstrate the advantages of our approach over the purely supervised setting.

## 2 Related Work

Research has demonstrated that an accurate semantic segmentation can not be achieved by only using pixel intensities of the images and it is essential to take into consideration the contextual information [7,8]. This contextual information can be defined as the relation between objects, *i.e.* sky above building, or the structure of an object itself, *i.e.* parts of the human body. Graphical models such as Conditional Random Fields (CRF) [9,10,11] and Markov Random Fields (MRF) [12,13,14,15] have been widely employed for semantic segmentation. CRF models consist of a unary term on each pixel or region of the image and a pairwise term that regulates consistency between neighboring pixels or regions. He *et al.* [11] use a CRF in a multi-scale approach to incorporate contextual information into the task of image labeling. Kohli *et al.* [10] use unsupervised methods to segment an image into regions and use a high order CRF to combine these regions. Accuracy of such methods is usually limited by the initial segmentation. To overcome this problem, CRFs have been used on the pixel level as well to produce more fine grained segmentation. Krahenbuhl and Koltun [9] define a fully connected CRF on the image and approximate the pairwise energy terms with Gaussian kernels. Gould *et al.* [13] utilize an MRF to define an energy function on both geometry and appearance of the image. They also introduce a two stage hill climbing inference technique to minimize this energy term. Tighe and Lazebnik [12] implement an MRF on top of image segments to optimize the context of neighborhood regions.

Deep learning has recently made significant advances in the field of computer vision. ConvNets [16] have been recently used for semantic segmenta-

tion [1,17,2,18,19,20,21,22]. Ciresan *et al.* [18] extracted patches around each pixel and trained a ConvNet to detect cell membranes in electron microscopy images. Farabet *et al.* [1] proposed a multi-scale ConvNet to extract features for scene labeling. Chen *et al.* [2] incorporate fully connected CRFs to improve semantic segmentation. Long *et al.* [17] propose a Fully Convolutional Network (FCN) which alters the fully connected layers in the network to convolutional layers. FCN takes an input image of any size and outputs a probability map of the same size. FCN has shown significant improvements in accuracy for dense predictions. Liu *et al.* [22] also use CRF on top of FCN and show fine grained improvements for semantic segmentation.

Cascaded classifiers have also been popular for the task of semantic segmentation [7,8,23,24,25,26,27]. These algorithms take advantage of multiple classifiers usually in a hierarchical or multi-scale fashion to boost the accuracy of the segmentation. Tu and Bai [8] introduced a series of classifiers that integrate both information from the original image and the contextual information from previous classifier in their auto-context algorithm. Seyedhosseini *et al.* [26,27] employ classifiers in a hierarchical fashion and incorporate contextual information from multiple scales. One significant difference between cascaded classifiers and ConvNets is that training cascaded classifiers is computationally easier due to the fact that each classifier is learned separately in an incremental fashion where ConvNets are trained in a unified manner and as one network. However, ConvNets are observed to produce slightly more accurate results than cascaded classifiers in general.

### 3 Unsupervised Loss Functions from Spatial Structure

Images have spatial characteristics that can be exploited for learning semantic segmentation. For instance, regions corresponding to distinct objects in natural images are expected to be piecewise smooth whereas vasculature in medical images is curvilinear (2D) or tubular (3D). While such knowledge has been extensively used for pre-processing input images or post-processing output images in the context of semantic segmentation, the contribution of our work is their use as unsupervised loss functions directly in learning. These loss functions are constructed using spatial operations on the output images of the classifier being learned. Therefore, multiple output image pixels are included in their computation compared to loss functions such as Mean Squared Error (MSE) where only one pixel is used to compute the error. Consequently, error backpropagation for the proposed loss functions are different. Next, we will show how these general spatial loss functions can be minimized using backpropagation. We will then derive the solution for the specific case of the total variation loss in Section 4.

Without loss of generality consider a binary scene labeling paradigm where the goal is to assign a label "zero" or "one" to each pixel in an image. We have a set of training images  $I_i \in \mathcal{I}$ . At the pixel level, the training set consists of two sets of pixels, (i) labeled pixels which in this paper we call "supervised" pixels,  $x_k \in \mathcal{S}$  and (ii) unlabeled pixels which we call "unsupervised" pixels,

$x_k \in \mathcal{U}$ . These two sets are disjoint and their union is the set of all pixels in  $\mathcal{I}$ . A supervised loss function such as mean squared error

$$E_S(\mathbf{w}) = \sum_{x_k \in \mathcal{S}} (f(x_k; \mathbf{w}) - L(x_k))^2, \quad (1)$$

where  $L(x_k)$  is the training label of pixel  $x_k$ , can be minimized to learn the parameters  $\mathbf{w}$  of the classifier  $f$ . However, when the number of free parameters  $\mathbf{w}$  is large and the amount of labeled data available is small in comparison, we can expect overfitting and poor performance. Of course, one solution is to increase the amount of labeled data; however, this is costly in human effort as discussed previously. Regularization by penalizing the  $L_2$  norm of the weight vector is another traditional approach to this problem; however, these types of regularization constraints are very general and do not exploit the characteristics of the spatial structures present in images.

Let  $P_i = f(\mathcal{I}_i)$  denote the probability map generated by the classifier for the training image  $\mathcal{I}_i$  and let  $\mathcal{P}$  denote the set of all such probability images for our training data. Let

$$E_{\mathcal{U}}(\mathbf{w}) = \sum_{x_k \in \mathcal{I}} \Theta(\bar{f}(x_k)) \quad (2)$$

where  $\bar{f}(x_k)$  is a vector of size  $n^2$  containing the  $n \times n$  neighborhood of pixel  $x_k$  in the probability map image  $P_i$  and  $\Theta : \mathbb{R}^{n^2} \rightarrow \mathbb{R}$  is a function which computes some spatial characteristic of the probability map image which we want to penalize. We should point out that labeled data can be treated as unlabeled data too, therefore we consider the whole set  $\mathcal{I}$  for defining the unsupervised loss. Consider an offset vector  $\gamma$  which defines the spatial offset of the  $n \times n$  neighborhood with respect to the  $k$ th pixel  $x_k$ . In this case we will have,

$$\bar{f}(x_k) = \begin{bmatrix} f_1(x_k) \\ f_2(x_k) \\ \dots \\ f_{n^2}(x_k) \end{bmatrix}_{n^2 \times 1} \quad (3)$$

where we have,

$$f_i(x_k) = f(x_{k+\gamma(i)}). \quad (4)$$

We note that  $E_{\mathcal{U}}$  is a function of  $f_i(x_k)$  and  $i = 1, 2, \dots, n^2$  which are themselves functions of  $\mathbf{w}$

$$E_{\mathcal{U}}(\mathbf{w}) = \sum_{x_k \in \mathcal{I}} \Theta(f_1(x_k; \mathbf{w}), f_2(x_k; \mathbf{w}), \dots, f_{n^2}(x_k; \mathbf{w})). \quad (5)$$

We adopt stochastic gradient descent to minimize this loss function [28,29]. By applying the multivariate chain rule to compute the gradient of  $E_{\mathcal{U}}$  with respect

to  $\mathbf{w}$  we have,

$$\frac{\partial E_{\mathcal{U}}}{\partial \mathbf{w}} = \sum_{x_k \in \mathcal{I}} \sum_{i=1}^{n^2} \frac{\partial E_{\mathcal{U}}}{\partial f_i(x_k)} \times \frac{df_i(x_k)}{d\mathbf{w}} \quad (6)$$

where  $\frac{df_i(x_k)}{d\mathbf{w}}$  is the derivative of  $f(x)$  with respect to  $\mathbf{w}$  computed at  $x = x_{k+\gamma(i)}$  and  $\frac{\partial E_{\mathcal{U}}}{\partial f_i(x_k)}$  is a coefficient which is defined by the specific choice of the unsupervised loss function. In other words, the backpropagation of a spatial loss function is a linear combination of the derivatives of  $f$  with respect to the weights calculated at specific locations with respect to a specific sample. The coefficients of this linear combination are dependent on the definition of the loss function. Finally, note that the proposed unsupervised loss can be combined with a supervised loss  $E_{\mathcal{S}} + \alpha E_{\mathcal{U}}$  to work in a semi-supervised learning setting.

## 4 Total Variation Loss for Learning

The piecewise smoothness property of natural images has been used extensively in the fields of image processing and computer vision. Perona and Malik [30] introduced a nonlinear PDE to remove noise from the image while conserving most details of the original image with the assumption that image intensities are piecewise flat. This diffusion results in piecewise homogeneous regions in the image which is achieved by controlling the diffusion around edges. Rudin *et al.* [31] propose to minimize the total variation of an image to suppress the noise and preserve the edges at the same time. We observe that natural scene image segmentation probability maps should result in piecewise smooth regions which means that if a pixel has a high probability of belonging to a specific object category, its neighboring pixels also have high probability to belong to the same category. This suggests that a probability map that generates an optimum segmentation of a natural scene image should have low total variation. This observation motivates us to use total variation of probability map images as an unsupervised penalty to help improve segmentation results when only sparsely labeled images are available. We define the unsupervised loss function as

$$E_{\mathcal{U}}(\mathbf{w}) = \sum_{x_k \in \mathcal{I}} \Theta(\bar{f}(x_k; \mathbf{w})) = \sum_{x_k \in \mathcal{I}} |\nabla f(x_k; \mathbf{w})| \quad (7)$$

$$= \sum_{x_k \in \mathcal{I}} |\nabla_X f(x_k)| + |\nabla_Y f(x_k)|. \quad (8)$$

Although equation (8) is convex, it is not differentiable. We use the subgradient method which generalizes the derivative to non-differentiable functions to compute

$$\frac{\partial E_{\mathcal{U}}}{\partial \mathbf{w}} = \sum_{x_k \in \mathcal{I}} \frac{d}{d\mathbf{w}} |\nabla_X f(x_k)| + \frac{d}{d\mathbf{w}} |\nabla_Y f(x_k)| \quad (9)$$

$$= \sum_{x_k \in \mathcal{I}} \text{sign}(\nabla_X f(x_k)) \frac{d}{d\mathbf{w}} \nabla_X f(x_k) + \text{sign}(\nabla_Y f(x_k)) \frac{d}{d\mathbf{w}} \nabla_Y f(x_k). \quad (10)$$

We use the Sobel operators

$$\nabla_X = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad \nabla_Y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (11)$$

to calculate the derivatives of the probability map image in  $X$  and  $Y$  directions, respectively. These derivative computations can be written as the vector products

$$\nabla_X f(x_k) = \overline{\nabla_X}^T \bar{f}(x_k), \quad \nabla_Y f(x_k) = \overline{\nabla_Y}^T \bar{f}(x_k), \quad (12)$$

where  $\overline{\nabla_X}$  and  $\overline{\nabla_Y}$  are the vectorized versions of  $\nabla_X$  and  $\nabla_Y$  operators in (11) in row major form, respectively, and the superscript  $T$  denotes vector transpose. Let  $\gamma(i)$ ,  $i = 1, \dots, 9$ , be the offsets in (4) chosen to represent the pixels in the  $3 \times 3$  neighborhood in a row major fashion beginning from top left and ending in bottom right. We then have

$$\nabla_X f(x_k) = \begin{bmatrix} -1 \\ -2 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} f_1(x_k) \\ f_2(x_k) \\ f_3(x_k) \\ f_4(x_k) \\ f_5(x_k) \\ f_6(x_k) \\ f_7(x_k) \\ f_8(x_k) \\ f_9(x_k) \end{bmatrix}, \quad \nabla_Y f(x_k) = \begin{bmatrix} -1 \\ 0 \\ 1 \\ -2 \\ 0 \\ 2 \\ -1 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} f_1(x_k) \\ f_2(x_k) \\ f_3(x_k) \\ f_4(x_k) \\ f_5(x_k) \\ f_6(x_k) \\ f_7(x_k) \\ f_8(x_k) \\ f_9(x_k) \end{bmatrix} \quad (13)$$

where  $f_i(x_k)$  is defined as in equation (4). Both spatial derivative operators  $\nabla_X$  and  $\nabla_Y$  and also the derivative operator  $\frac{d}{d\mathbf{w}}$  are linear and interchangeable. Interchanging the order of differentiation with respect to  $\mathbf{w}$  and the differentiation with respect to  $X$  and  $Y$  in (10), and substituting the expressions for the spatial derivatives into (10), we obtain

$$\begin{aligned} \frac{\partial E_U}{\partial \mathbf{w}} = & \sum_{x_k \in \mathcal{I}} \sum_{i=1}^9 \text{sign}(\nabla_X f(x_k)) \overline{\nabla_X}(i) \frac{df_i(x_k)}{d\mathbf{w}} \\ & + \text{sign}(\nabla_Y f(x_k)) \overline{\nabla_Y}(i) \frac{df_i(x_k)}{d\mathbf{w}} \end{aligned} \quad (14)$$

where  $\overline{\nabla_X}(i)$  and  $\overline{\nabla_Y}(i)$  are the  $i$ th element in vectors  $\overline{\nabla_X}$  and  $\overline{\nabla_Y}$ , respectively. In the last section we derived the gradient for a general spatial penalty on the probability map image and obtained equation (6). Comparing equation (6) with equation (14) which shows the gradient of the total variation penalty we can observe that

$$\frac{\partial E_U}{\partial f_i(x_k)} = \text{sign}(\nabla_X f(x_k)) \times \overline{\nabla_X}(i) + \text{sign}(\nabla_Y f(x_k)) \times \overline{\nabla_Y}(i) \quad (15)$$

The gradient of the unsupervised penalty at  $x_k$  is defined by a linear combination of gradient values of the system at  $x_k$ 's eight connected neighborhood pixels. This implies a dependency between pixels in the probability map image which is the goal of total variation penalty.

## 5 Experiments

In this section, we use two different pixel classifiers, Contextual Hierarchical Model (CHM) [27] and ConvNet pixel classifier [18], to show that the proposed loss function can improve accuracy regardless of the specific classifier choice. CHM [27] is a cascaded pixel classifier which combines multiple classifiers in a hierarchical fashion to boost its accuracy. CHM consists of two stages, bottom-up and top-down. In the bottom-up stage CHM creates a multi-scale pyramid of contextual maps which are generated by applying a classifier at each scale on the downsampled original image and the previous contextual maps in the pyramid. As we go to higher levels in the bottom-up stage, the resolution of the image decreases which means the higher level classifier is able to capture more global features than a lower level classifier which captures more fine-grained details. Finally, all the contextual maps in the pyramid created at the bottom-up stage and the image features from the original image are used in the top-down stage classifier to compute the final probability map image. We use CHM as our learning system for experiments on the Weizmann horse dataset.

We also employ the ConvNet proposed in [18] for pixel classification, with some modifications, to show practicality of our unsupervised penalty on ConvNets. In this model we extract a patch around each pixel of the image and train a ConvNet to classify that pixel. In the original work by Ciresan et al, multiple data augmentations and averaging of different models are used to get state of the art results. Since the goal of this paper is not achieving state of the art results, but to show improvements in a semi-supervised paradigm we have not performed any data augmentation or classifier averaging. We used a ConvNet consisting of 4 convolutional layers containing 64 maps and a kernel size of  $3 \times 3$ , each one followed by a ReLU layer, a maxpooling layer and a Local Response Normalization (LRN) layer and is continued with a fully connected layer of size 512. Finally, a softmax layer is used to generate probabilities for each class in the dataset. We used Berkeley Vision and Learning Center's Caffe [32] framework to implement this ConvNet.

### 5.1 Weizmann Horse Data Set

The Weizmann Horse data set [33] consists of 328 gray scale images of horses with different sizes provided with pixel labels for background and foreground. We followed the approach by [8,27] and use half of the images as training and the other half for testing. Note that we are considering a setting where only sparse labels are available. We randomly choose a specific number of labeled pixels from each training image and consider the rest of the pixels as unlabeled training data

to create sparse label sets for training. We randomly created sparse label training sets of sizes 10, 20, 30, 40, and 50 labeled pixels per training image. We generate five different sparse label training sets for each size and report the average and standard deviation on the testing data.

For each of these sets we trained two different CHMs with 3 levels in the bottom-up stage. One CHM was trained with the supervised penalty and the other one with both the supervised penalty and unsupervised penalty. Note that we only applied the unsupervised penalty to the classifiers in CHM that work on the original resolution of the image and not the classifiers that have lower resolution. We applied the penalty to both the first classifier of the bottom-up stage and the classifier of the top-down stage. The reasoning behind this is the fact that CHM tends to generate smooth probability maps in coarser resolutions inherently and applying the total variation to those levels does not improve the results significantly.

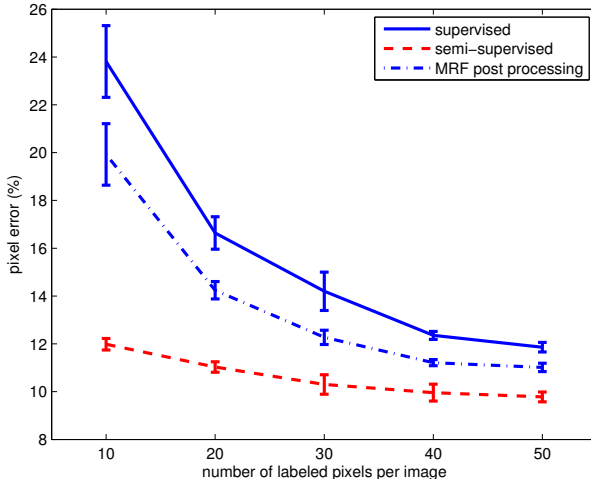
**Table 1.** Pixel error results for Weizmann Horse dataset. The reported numbers are average of 5 experiments and corresponding standard deviation. Average(%) $\pm$  std dev.

# of labeled pixels per image (% of whole training set)	supervised approach	MRF post processing	semi-supervised approach
10 (0.0168 %)	23.81 $\pm$ 1.50	19.92 $\pm$ 1.28	11.98 $\pm$ 0.24
20 (0.0336 %)	16.63 $\pm$ 0.67	14.24 $\pm$ 0.36	11.03 $\pm$ 0.21
30 (0.0504 %)	14.20 $\pm$ 0.80	12.27 $\pm$ 0.29	10.30 $\pm$ 0.41
40 (0.0671 %)	12.35 $\pm$ 0.16	11.21 $\pm$ 0.12	9.96 $\pm$ 0.35
50 (0.0839 %)	11.86 $\pm$ 0.19	11.01 $\pm$ 0.17	9.78 $\pm$ 0.20

The results on the testing dataset can be seen in Table (1), Figures (1) and (2). The total variation loss we proposed for learning improves the results significantly, especially when we have very sparse labeled data available. It is noteworthy that the CHM used for this experiment achieves a pixel error of 5.73% when fully labeled data is used to train the model. The state of the art pixel error on this dataset is 4.1% [34]. Therefore, somewhat better results can be expected if our proposed unsupervised loss function is applied to the model in [34]. We also note that using the proposed total variation loss does not significantly effect the results when the full set of pixels in the training set is treated as the supervised set. This is expected as the characteristics of spatial structures in the training images can be learned directly from the fully labeled data in this case.

Since the proposed unsupervised loss function can be seen as a regularization term in learning, we compared to classical Tikhonov regularization by penalizing the  $L_2$  norm of the weight vector  $\mathbf{w}$ . However, the results obtained with this approach were not significantly better than the purely supervised learning without regularization.





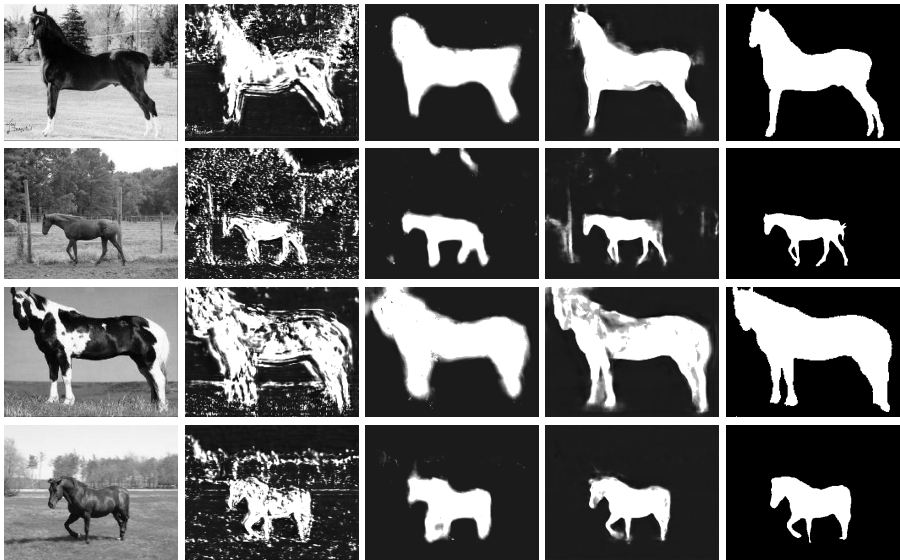
**Fig. 1.** Mean testing error plot for the Weizmann Horse dataset. Bars indicate standard deviation over the five trials at each size.

Finally, we also demonstrate the advantages of imposing spatial constraints in the learning phase vs. the inference phase. Therefore, we used the UGM framework [35] to implement an MRF that was used to post-process the resulting probability images from the supervised only approach. Although the accuracy was increased by MRF post-processing (see Table 1), the proposed total variation loss has much lower error rates.

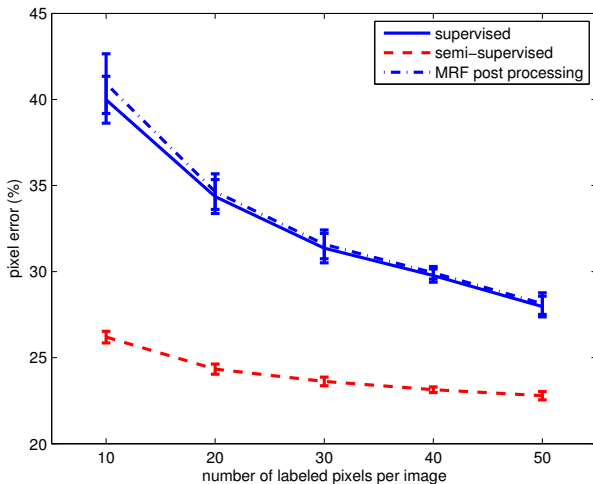
## 5.2 Stanford Background Data Set

The Stanford Background dataset [13] contains 715 RGB images and the corresponding label images. Images are approximately  $240 \times 320$  pixels in size and pixels are classified into eight different categories. We follow the standard evaluation procedure for this dataset and do a five-fold cross validation. We randomly divide the images into 5 separate bins, each containing 143 images. We train on 4 of these bins and test on the other one and repeat the procedure for all 5 bins.

We randomly created sparse label training sets of sizes 10, 20, 30, 40, and 50 labeled pixels per training image. We generate five different sparse label training sets for each size and report the average and standard deviation on the testing data. We use the ConvNet explained in the beginning of Section 5 as the pixel classifier in this experiment. The mean testing pixel errors and standard deviations can be seen in Table (2) and Figure (3). We post processed results of purely supervised learning with a simple MRF, and as can be observed from Table (2) and Figure (3) there is not significant improvement in the results. The supervised learning results are too noisy and even in some cases incorrect that an MRF cannot make significant corrections. Note that training better MRF and



**Fig. 2.** Sample images from Weizmann Horse dataset, using only 10 labeled pixel per image for training. The first column shows the original image, the second column are the output of the purely supervised method and the third column images are the results of the proposed method. The fourth column shows the probability maps of the model using dense (full) labels for training and the last column shows the ground truth images.

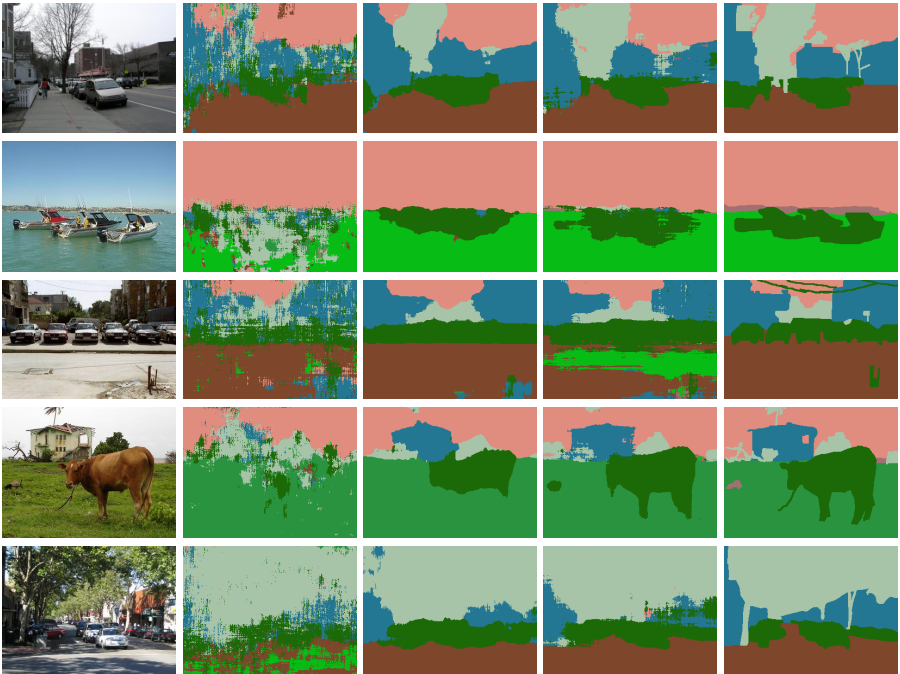


**Fig. 3.** Mean testing error plot for the Stanford background dataset. Bars indicate standard deviation over the five trials at each size.

**Table 2.** Pixel error results for Stanford Background dataset. The reported numbers are average of 5 experiments and corresponding standard deviation. Average(%) $\pm$  std dev.

# of labeled pixels per image (% of whole training set)	supervised approach	MRF post processing	semi-supervised approach
10 (0.0130 %)	$39.98 \pm 1.35$	$40.92 \pm 1.73$	$26.19 \pm 0.33$
20 (0.0260 %)	$34.35 \pm 0.98$	$34.64 \pm 1.03$	$24.33 \pm 0.29$
30 (0.0391 %)	$31.36 \pm 0.85$	$31.58 \pm 0.83$	$23.61 \pm 0.25$
40 (0.0521 %)	$29.76 \pm 0.38$	$29.93 \pm 0.36$	$23.13 \pm 0.16$
50 (0.0651 %)	$27.96 \pm 0.60$	$28.13 \pm 0.62$	$22.79 \pm 0.24$

CRF models also is contingent on availability of dense labeled training images. An important advantage of our method is its ability to work in a sparsely labeled training image paradigm.



**Fig. 4.** Sample images from Stanford Background dataset, using only 10 labeled pixel per image for training. The first column shows the original image, the second column are the output of the purely supervised method and the third column images are the results of the proposed method. The fourth column shows the probability maps of the model using dense (full) labels for training and the last column shows the ground truth images.

For comparison, we also trained a ConvNet where we use full labels of the dataset. We achieved a pixel error of 20.55% after one epoch over all the pixels in the training set, where the state of the art pixel error for this data set [27] is 17.05%. Visual examples are given in Figure (4). Note that while the purely supervised learning results in a classifier whose output label maps are noisy whereas the proposed approach results in a classifier which generates piecewise smooth labeled regions and is closer visually to supervised learning with the dense label set.

### 5.3 Sift Flow Data Set

The Sift Flow dataset [36] contains 2488 training images and 200 testing images all of size  $256 \times 256$ . It contains 33 different classes of which only 30 of them are present in the test set. We use the same ConvNet used for evaluating the Stanford Background dataset as our pixel classifier. We follow the same procedure as before to generate sparse label sets. The results for the Sift Flow data set are given in Table (3) and Figure (5). We also post processed supervised results with an MRF to increase consistency but it did not make any significant changes in the results. Note that the MRFs we used on these datasets to improve label consistency are not trained with any dense labels since we are considering a setting where only sparse labels are available. Therefore, it is expected that MRFs will not result in significant improvement, especially in the multi-class cases where dense label information can be more informative to the MRF.

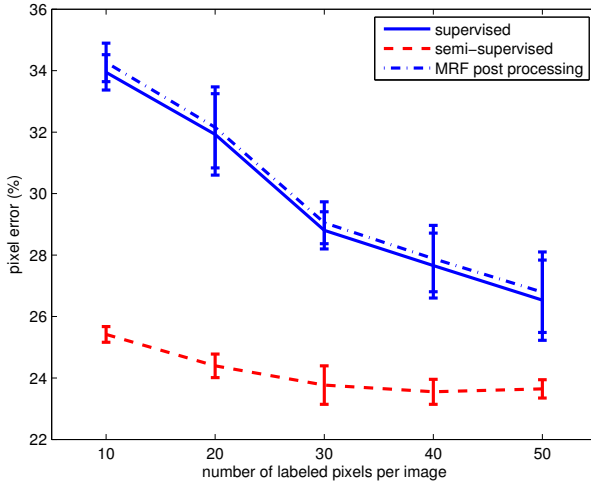
We also trained our model using dense labels where we achieved a pixel error of 20.78%. The state of the art pixel error on this data set before the significant improvement of Long *et al.* [17] to a 14.8% pixel error was achieved by Tighe *et al.* [37] with a pixel error of 21.4%.

**Table 3.** Pixel error results for Sift Flow dataset. The reported numbers are average of 5 experiments and corresponding standard deviation. Average(%) $\pm$  std dev.

# of labeled pixels per image (% of whole training set)	supervised approach	MRF post processing	semi-supervised approach
10 (0.0153 %)	$33.94 \pm 0.57$	$34.26 \pm 0.62$	$25.41 \pm 0.25$
20 (0.0305 %)	$31.92 \pm 1.32$	$32.15 \pm 1.31$	$24.39 \pm 0.38$
30 (0.0458 %)	$28.80 \pm 0.60$	$29.05 \pm 0.68$	$23.77 \pm 0.62$
40 (0.0610 %)	$27.66 \pm 1.05$	$27.88 \pm 1.08$	$23.55 \pm 0.40$
50 (0.0763 %)	$26.53 \pm 1.30$	$26.79 \pm 1.31$	$23.64 \pm 0.29$

## 6 Conclusion

In this paper we proposed to enforce spatial constraints on the output probability maps of learning algorithms. These spatial constraints help the learning system in converging to a better minimum in the solution space, especially in the case when we do not have enough labeled data. We proposed and showed how to

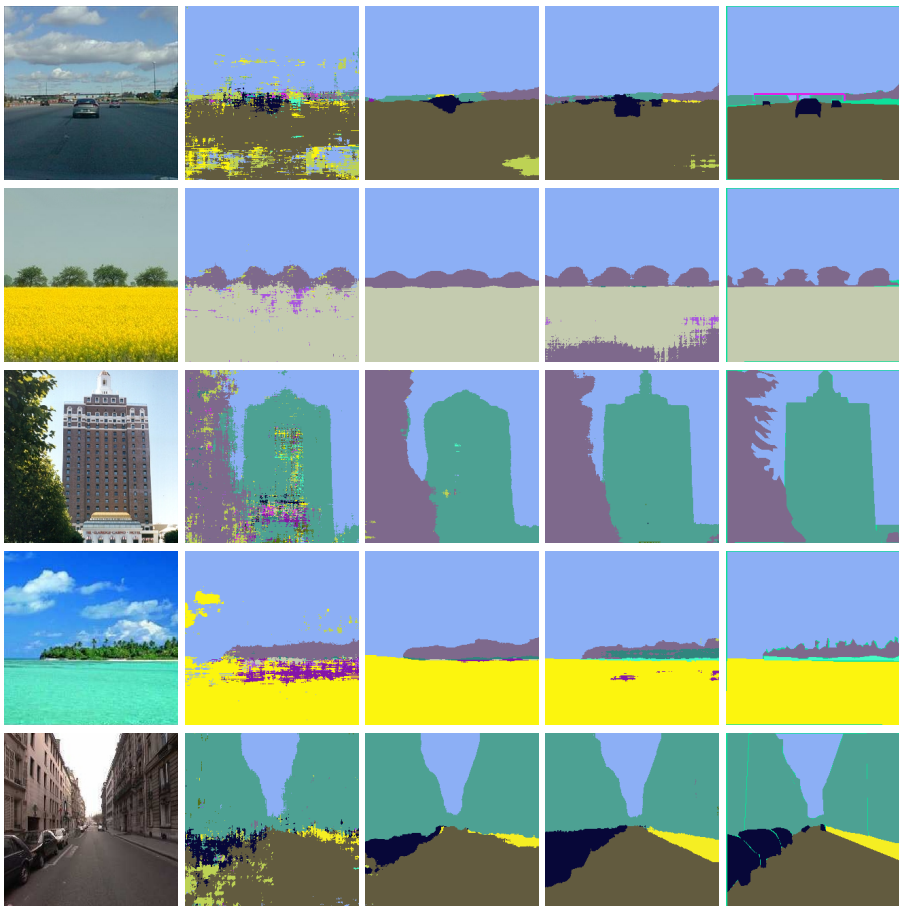


**Fig. 5.** Mean testing error plot for the Sift Flow dataset. Bars indicate standard deviation over the five trials at each size.

minimize these constraint through learning via backpropagation in any pixel classifier. Our approach can be useful for learning deep structures like ConvNets, since deep structures need large amount of training data for a good performance. The proposed method uses the sparse labeled data to avoid any trivial solution to the unsupervised loss function. From the probabilistic point of view we can interpret the spatial constraint proposed in this paper as a prior probability on the learning system parameters that restrict the solution to some smooth functions. Considering a graphical model paradigm, we can interpret the spatial constraints as the pairwise energy term in classical CRF models, where the unary energy term represents the supervised energy.

We showed significant improvement on three different datasets and two different pixel classifiers when only sparse labels are available. It is noteworthy that focus of this paper is to show practicality and effectiveness of the spatial constraints for sparse labels, not proposing an algorithm to achieve state of the art pixel accuracy on the task of semantic segmentation. We should point out the results reported in this paper can be further improved by selecting better models or selecting sparse labels by an expert who can help the learning algorithm by selecting informative pixels to label rather than randomly labeling pixels, as we have done in the paper.

Finally, we note that our framework is general and can be used to construct other unsupervised loss functions based on spatial characteristics of different images. For instance, substantial work has been carried out to enhance curvilinear and tubular structures in classical image processing. These methods typically make use of the first-order local derivative structure, i.e. structure tensor [38], or



**Fig. 6.** Sample images from Sift Flow dataset, using only 10 labeled pixel per image for training. The first column shows the original image, the second column are the output of the purely supervised method and the third column images are the results of the proposed method. The fourth column shows the probability maps of the model using dense (full) labels for training and the last column shows the ground truth images.

second-order local derivative structure, i.e. Hessian [39,40] of images containing curvilinear and tubular structures. Both of these approaches can be formulated as unsupervised loss functions for learning following the general methodology introduced in Section 3.

## References

1. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **35**(8) (2013) 1915–1929

2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)
3. Zhu, X., Vondrick, C., Ramanan, D., Fowlkes, C.: Do we need more training data or better models for object detection?. In: BMVC. Volume 3., Citeseer (2012) 5
4. Chapelle, O., Schölkopf, B., Zien, A., et al.: Semi-supervised learning. (2006)
5. Zhu, X.: Semi-supervised learning literature survey. (2005)
6. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, ACM (1998) 92–100
7. Torralba, A., Murphy, K.P., Freeman, W.T.: Contextual models for object detection using boosted random fields. In: Advances in neural information processing systems. (2004) 1401–1408
8. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(10) (2010) 1744–1757
9. Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Adv. Neural Inf. Process. Syst (2011)
10. Kohli, P., Torr, P.H., et al.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision **82**(3) (2009) 302–324
11. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on. Volume 2., IEEE (2004) II–695
12. Tighe, J., Lazebnik, S.: Superparsing. International Journal of Computer Vision **101**(2) (2013) 329–349
13. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 1–8
14. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–7
15. Kumar, M.P., Koller, D.: Efficiently selecting regions for scene understanding. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 3217–3224
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
18. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in neural information processing systems. (2012) 2843–2851
19. Pinheiro, P.H., Collobert, R.: Recurrent convolutional neural networks for scene parsing. arXiv preprint arXiv:1306.2795 (2013)
20. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Computer vision–ECCV 2014. Springer (2014) 297–312
21. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation: Supplementary material (2014)

22. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1377–1385
23. Fink, M., Perona, P.: Mutual boosting for contextual inference. In: Advances in neural information processing systems. (2003) None
24. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: Advances in Neural Information Processing Systems. (2009) 641–648
25. Li, C., Kowdle, A., Saxena, A., Chen, T.: Towards holistic scene understanding: Feedback enabled cascaded classification models. In: Advances in Neural Information Processing Systems. (2010) 1351–1359
26. Seyedhosseini, M., Sajjadi, M., Tasdizen, T.: Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2168–2175
27. Seyedhosseini, M., Tasdizen, T.: Semantic image segmentation with contextual hierarchical models. (2015)
28. Bottou, L.: Stochastic gradient learning in neural networks. Proceedings of Neuro-Nimes **91**(8) (1991)
29. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.R.: Efficient backprop. In: Neural networks: Tricks of the trade. Springer (2012) 9–48
30. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on **12**(7) (1990) 629–639
31. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1) (1992) 259–268
32. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM (2014) 675–678
33. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(12) (2008) 2109–2125
34. Liu, G., Lin, Z., Tang, X., Yu, Y.: A hybrid graph model for unsupervised object segmentation. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE (2007) 1–8
35. Schmidt, M.: Ugm: Matlab code for undirected graphical models (2012)
36. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing via label transfer. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(12) (2011) 2368–2382
37. Tighe, J., Lazebnik, S.: Finding things: Image parsing with regions and per-exemplar detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3001–3008
38. Weickert, J.: Coherence-enhancing diffusion filtering. International Journal of Computer Vision **31**(2-3) (1999) 111–127
39. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: Medical Image Computing and Computer-Assisted InterventionMICCAI98. Springer (1998) 130–137
40. Tasdizen, T., Whitaker, R., Marc, R., Jones, B.: Enhancement of cell boundaries in transmission electron microscopy images. In: Image Processing, 2005. ICIIP 2005. IEEE International Conference on. Volume 2., IEEE (2005) II–129